

# Type-logical grammar in Agda

Pepijn Kokke

Utrecht University

August 6th 2015

# Take-home message

Machine-checked linguistic theories, which are directly embedded in a published work, are within reach.

(Sometimes a little bit of typesetting is nice, though.)

# Example

$ex_1$  : ✓ mary finds a unicorn

$ex_1 = \_$

$ex_2$  : ✓ ( a unicorn ) finds mary

$ex_2 = \_$

$ex_3$  : \* unicorn unicorn unicorn unicorn

$ex_3 = \_$

(The constructors of the parse tree have been omitted, as they are superfluous.)

# Example

```
isPreorder : IsPreorder _≡_ _⊢NL_  
isPreorder = record  
  { isEquivalence = ≡.isEquivalence  
  ; reflexive      = ax'  
  ; trans         = cut'  
  }
```

(I realise this doesn't say much at the moment, but we're getting there.)

# Types and judgement

data Type : Set where

el : Atom → Type

\_ $\otimes$ \_ : Type → Type → Type

\_ $\setminus$ \_ : Type → Type → Type

\_ $\diagup$ \_ : Type → Type → Type

data Judgement : Set where

\_ $\vdash$ \_ : Type → Type → Judgement

# The non-associative Lambek calculus

data NL : Judgement  $\rightarrow$  Set where

ax :  $e \vdash e$

$m \otimes$  :  $A \vdash B \rightarrow C \vdash D \rightarrow A \otimes C \vdash B \otimes D$

$m \setminus$  :  $A \vdash B \rightarrow C \vdash D \rightarrow B \setminus C \vdash A \setminus D$

$m /$  :  $A \vdash B \rightarrow C \vdash D \rightarrow A / D \vdash B / C$

$r \setminus \otimes$  :  $B \vdash A \setminus C \rightarrow A \otimes B \vdash C$

$r \otimes \setminus$  :  $A \otimes B \vdash C \rightarrow B \vdash A \setminus C$

$r / \otimes$  :  $A \vdash C / B \rightarrow A \otimes B \vdash C$

$r \otimes /$  :  $A \otimes B \vdash C \rightarrow A \vdash C / B$

(Each judgement should be prefixed with NL, but in the interest of readability we will use  $A \vdash B = \text{NL } A \vdash B$ .)

# Identity expansion

$ax' : A \vdash A$

$ax' \{A = \text{el } A\} = ax$

$ax' \{A = A \otimes B\} = m \otimes ax' ax'$

$ax' \{A = A \wedge B\} = m \wedge ax' ax'$

$ax' \{A = A \setminus B\} = m \setminus ax' ax'$

( $\{A=...\}$  is Agda syntax to match on implicit parameters.)

# Cut elimination

- All connectives are introduced by monotonicity rules.
- Each connective can be affected by residuation at the top-level on only *one* side of the turnstile.
- In a cut, we have the top-level connective available on *both* sides.
- So: we can be sure to find the monotonicity rule which introduced the connective in one of the arguments.



# Cut elimination

For instance, in the case of  $\otimes$ :

$$\frac{\frac{\frac{E \vdash B \quad F \vdash C}{E \otimes F \vdash B \otimes C}}{\vdots} \quad \frac{A \vdash B \otimes C \quad B \otimes C \vdash D}{A \vdash D}}{\frac{E \vdash B \quad \frac{\frac{F \vdash C \quad \frac{\frac{B \otimes C \vdash D}{C \vdash B \setminus D}}{F \vdash B \setminus D}}{B \otimes F \vdash D}}{B \vdash D \setminus F}}{E \vdash D \setminus F}}{E \otimes F \vdash D}}{\vdots} \quad A \vdash D} \rightsquigarrow$$

# Origins

$$\frac{\frac{E \vdash B \quad F \vdash C}{E \otimes F \vdash B \otimes C}}{\vdots} \frac{}{A \vdash B \otimes C}$$

# Origins

$$\frac{\frac{h_1 : E \vdash B \quad h_2 : F \vdash C}{m \otimes h_1 h_2 : E \otimes F \vdash B \otimes C}}{f : \vdots} \frac{f (m \otimes h_1 h_2) : A \vdash B \otimes C}{}$$

# Origins

```
data Origin'  
  ( f : A ⊢ B ⊗ C )  
  : Set where
```

```
origin : ( h1 : E ⊢ B  
         → ( h2 : F ⊢ C  
           → ( f' : E ⊗ F ⊢ G → A ⊢ G )  
           → ( pr : f ≡ f' ( m ⊗ h1 h2 ) )  
           → Origin' f
```

(Function  $f'$  should work for *any* type  $G$ .)

# Origins

$$\frac{\frac{E \vdash B \quad F \vdash C}{E \otimes F \vdash B \otimes C}}{\vdots}$$
$$\frac{\quad}{A \otimes D \vdash B \otimes C}$$

# Origins

$$\frac{\frac{E \vdash B \quad F \vdash C}{E \otimes F \vdash B \otimes C}}{\vdots} \frac{A \vdash (B \otimes C) \swarrow D}{A \otimes D \vdash B \otimes C}$$

# Contexts

```
data Polarity : Set where + - : Polarity
```

```
data Context (p : Polarity) : Polarity → Set where
```

```
  [] : Context p p
```

```
  _ ⊗ > _ : Type → Context p + → Context p +
```

```
  _ \ > _ : Type → Context p - → Context p -
```

```
  _ / > _ : Type → Context p + → Context p -
```

```
  _ < ⊗ _ : Context p + → Type → Context p +
```

```
  _ < \ _ : Context p + → Type → Context p -
```

```
  _ < / _ : Context p - → Type → Context p -
```

# Contexts

$\_ [ \_ ] : \text{Context } p_1 p_2 \rightarrow \text{Type} \rightarrow \text{Type}$

$[ ] \quad [ A ] = A$

$B \otimes > C \quad [ A ] = B \otimes (C [ A ])$

$B \setminus > C \quad [ A ] = B \setminus (C [ A ])$

$B / > C \quad [ A ] = B / (C [ A ])$

$C < \otimes B \quad [ A ] = (C [ A ]) \otimes B$

$C < \setminus B \quad [ A ] = (C [ A ]) \setminus B$

$C < / B \quad [ A ] = (C [ A ]) / B$



# Contexts

```
data ContextJ (p : Polarity) : Set where
  _<⊢_ : Context p + → Type           → ContextJ p
  _⊢>_ : Type                          → Context p - → ContextJ p
```

```
_[_]J : ContextJ p → Type → Judgement
A <⊢ B [ C ]J = A [ C ] ⊢ B
A ⊢> B [ C ]J = A ⊢ B [ C ]
```

# Origins (revisited)

data Origin

(  $J : \text{Context}^J \text{ -- } )$   
(  $f : \text{NL } J [ B \otimes C ]^J$  )  
: Set where

origin : (  $h_1 : E \vdash B$  )  
→ (  $h_2 : F \vdash C$  )  
→ (  $f' : E \otimes F \vdash G \rightarrow \text{NL } J [ G ]^J$  )  
→ (  $pr : f \equiv f' (m \otimes h_1 h_2)$  )  
→ Origin J f

# Origins (revisited)

```
view : ( J : ContextJ - ) ( f : NL J [ B ⊗ C ]J ) → Origin J f
view ( . _ ⊢> [] ) ( m ⊗ f g ) = origin f g id refl
view ( . _ ⊢> [] ) ( r \ ⊗ f ) = wrap r \ ⊗ f
view ( . _ ⊢> [] ) ( r / ⊗ f ) = wrap r / ⊗ f
      ⋮
```

# Origins (revisited)

`view` : (  $J$  : `Context` <sup>$J$</sup>   $-$  ) (  $f$  : `NL`  $J$  [  $B \otimes C$  ] <sup>$J$</sup>  )  $\rightarrow$  `Origin`  $J$   $f$

`view` (  $\_$   $\vdash$   $\_$  [ ] ) (  $m \otimes f$   $g$  ) = `origin`  $f$   $g$  `id` `refl`

`view` (  $\_$   $\vdash$   $\_$  [ ] ) (  $r \setminus \otimes f$  ) = `wrap`  $r \setminus \otimes f$

`view` (  $\_$   $\vdash$   $\_$  [ ] ) (  $r / \otimes f$  ) = `wrap`  $r / \otimes f$

$\vdots$

`wrap` : (  $g$  : `NL`  $I$  [  $G$  ] <sup>$J$</sup>   $\rightarrow$  `NL`  $J$  [  $G$  ] <sup>$J$</sup>  ) (  $f$  : `NL`  $I$  [  $B \otimes C$  ] <sup>$J$</sup>  )  
 $\rightarrow$  `Origin`  $J$  (  $g$   $f$  )

`wrap`  $g$   $f$  `with` `view`  $I$   $f$

`wrap`  $g$   $f$  | `origin`  $h_1$   $h_2$   $f'$   $pr$  = `origin`  $h_1$   $h_2$  (  $g \circ f'$  ) ( `cong`  $g$   $pr$  )

(A `with` statement is a way to pattern match on the result of a function.)

# Cut elimination (revisited)

$\text{cut}' : A \vdash B \rightarrow B \vdash C \rightarrow A \vdash C$

$\text{cut}' \{B = \text{el } \_ \} f g \text{ with } \text{el.view } ([ ] <\vdash \_ ) g$   
... |  $\text{el.origin } g' \_ = g' f$

$\text{cut}' \{B = \_ \otimes \_ \} f g \text{ with } \otimes.\text{view } (\_ \vdash > [ ] ) f$   
... |  $\otimes.\text{origin } h_1 h_2 f \_ =$   
 $f (r \swarrow \otimes (\text{cut}' h_1 (r \otimes \swarrow (r \searrow \otimes (\text{cut}' h_2 (r \otimes \searrow g))))))$   
:

# Cut elimination (revisited)

$\text{cut}' : A \vdash B \rightarrow B \vdash C \rightarrow A \vdash C$

$\text{cut}' \{B = \text{el } \_ \} f g \text{ with } \text{el.view } ([ ] \text{ < } \vdash \_ ) g$

... |  $\text{el.origin } g' \_ = g' f$

$\text{cut}' \{B = \_ \otimes \_ \} f g \text{ with } \otimes.\text{view } (\_ \vdash > [ ] ) f$

... |  $\otimes.\text{origin } h_1 h_2 f \_ =$

$f (r / \otimes (\text{cut}' h_1 (r \otimes / (r \backslash \otimes (\text{cut}' h_2 (r \otimes \backslash g))))))$

$\text{cut}' \{B = \_ \backslash \_ \} f g \text{ with } \backslash.\text{view } ([ ] \text{ < } \vdash \_ ) g$

... |  $\backslash.\text{origin } h_1 h_2 g' \_ =$

$g' (r \otimes \backslash (r / \otimes (\text{cut}' h_1 (r \otimes / (\text{cut}' (r \backslash \otimes f) h_2))))$

$\text{cut}' \{B = \_ / \_ \} f g \text{ with } /.\text{view } ([ ] \text{ < } \vdash \_ ) g$

... |  $/.\text{origin } h_1 h_2 g' \_ =$

$g' (r \otimes / (r \backslash \otimes (\text{cut}' h_2 (r \otimes \backslash (\text{cut}' (r / \otimes f) h_1))))$

Utrechts  
**Universiteitsfonds**



**Universiteit Utrecht**